



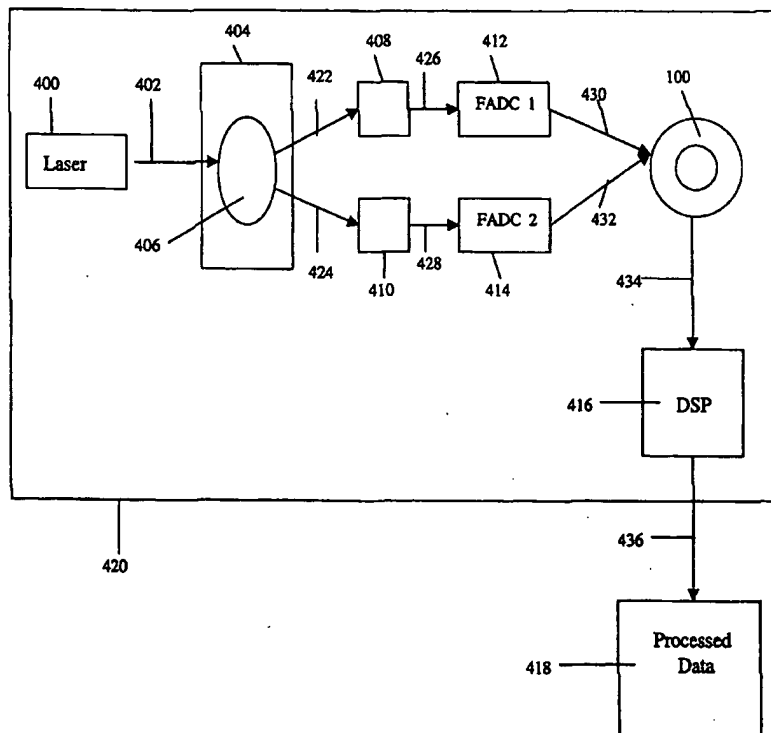
## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification :  Not classified		A2	(11) International Publication Number: <b>WO 99/57955</b>  (43) International Publication Date: 18 November 1999 (18.11.99)
(21) International Application Number: PCT/US99/10318 (22) International Filing Date: 13 May 1999 (13.05.99) (30) Priority Data: 60/085,381           14 May 1998 (14.05.98)   US 60/085,415           14 May 1998 (14.05.98)   US (71) Applicant: LUMINEX CORPORATION [US/US]; 12212 Technology Boulevard #K, Austin, TX 78727-6115 (US). (72) Inventor: CHANDLER, Van, S.; 7300 Valburn Drive, Austin, TX 78731 (US). (74) Agents: KOSHY, Suresh et al.; Pepper Hamilton LLP, 600 Fourteenth Street, N.W., Washington, DC 20005-2004 (US).		(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).  <b>Published</b> <i>With international search report.</i> <i>With amended claims.</i>	

(54) Title: ZERO DEAD TIME ARCHITECTURE AND METHOD FOR FLOW CYTOMETER

## (57) Abstract

A method and apparatus for providing a zero dead time for flow analyzers, flow cytometers, and other measurement devices. A circular buffer is used to store data from a flow analyzer in a plurality of data storage areas, until it is convenient to implement more time consuming data processing procedures. User specified parameters, including sampling rate and/or sampling period, size and number of data storage areas, size of the circular or other buffer, signal-to-noise threshold, order of processing when a plurality of Digital Signal Processors (DSPs) is used, and fixed trailing distance, are used to provide flexible and convenient operation by a user. The probability of missing a rare event occurring within the laser beam or other light beam of a flow analyzer or other measurement device is reduced to zero. Functional equivalents may be used in place of the circular buffer, for example, First In-First Out buffers which route data to a cache for potential re-reading and/or reprocessing or additional processing of data, cascading buffers, etc. Circular and functionally equivalent buffers permit sophisticated data processing in which preceding data (for example, data not meeting the S/N threshold) subsequently are re-analyzed for the possibility that the data are part of a developing event peak, such as a developing shoulder thereof, or part of a declining peak. In addition, logic and algorithms to permit re-examination and potential reprocessing of "old" subthreshold data is utilized to improve the S/N ratio of an event peak.



**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

5

**ZERO DEAD TIME ARCHITECTURE AND METHOD FOR FLOW CYTOMETER**

10

**FIELD OF THE INVENTION**

15       The present invention relates generally to an apparatus for, and a method of, increasing the quantity of information obtained from flow analyzers or other data collection devices by temporarily storing data relating to, for example, particles, cells, and inter-event background noise from flow through an examination zone. In particular, the present invention relates to an apparatus for and a method of achieving zero dead time by temporarily storing such data in a circular buffer, or similar buffer or temporary storage capability, for later  
20       processing.

**BACKGROUND OF THE INVENTION**

Flow cytometry is conveniently used to quickly measure one or more optical parameters of the cells and/or particles that pass through a light beam that impinges on a narrow examination zone. Background  
25       information on flow cytometry is found, for example, in Shapiro's Practical Flow Cytometry, Third Edition (Alan R. Liss, Inc. 1995), incorporated herein by reference in its entirety.

In conventional flow cytometers, sample fluid containing sample cells and/or particles is introduced from a sample tube into or near the center of a faster flowing stream of sheath fluid, which draws the sample toward the center of the combined streams. This process, known as hydrodynamic focusing, allows the cells  
30       and/or particles to be delivered reproducibly to the center of the measuring point in the flow cell or other examination zone. Typically, the cells and/or particles are in suspension in the flow cell.

5       A continuous wave laser focuses a laser beam on the cells and/or particles as they pass through the examination zone by a flow of a stream of the suspension. When an object of interest in the flow stream is struck by the laser beam, certain signals result and are sensed by detectors. For example, these signals include forward light scatter intensity, which provides information concerning the size of individual cells and/or particles, and side light scatter intensity, which provides information regarding the relative size and refractive  
10       properties of individual cells and/or particles.

Other signals which may be sensed by detectors include fluorescence emissions from one or more fluorescent dyes and/or other fluorescent molecules, for example, tryptophan or other fluorescent amino acid(s) or other molecule(s) that is native to a protein or other peptide, or to another biomolecule or man-made molecule. Typically, when a plurality of different fluorescing molecules is employed in an analytical scheme,  
15       fluorescence emission peaks are selected to minimize or, ideally, eliminate spectral overlap between the respective fluorescence emission peaks.

For example, ideally, fluorescence emission peaks will differ by 50  $\mu\text{m}$ , though lesser (or larger) spectral separation also can be accommodated and used to advantage, for example, 20, 30 or 40  $\mu\text{m}$ , where the greater the spectral separation, the more powerful is the discrimination between the respective fluorescence emitters. In  
20       addition, quantum efficiency is considered in the choice of fluorescent molecules. In the case of use of a plurality of different fluorescent molecules, a separate detector is tuned to and used for the different wavelength of emission of each fluorescence emitter. The excitation wavelength for more than one fluorescent molecule can be the same, or different excitation wave lengths can be used to match the excitation spectrum of each different fluorescent molecule.

25       The optical signals that are generated by an analytical procedure are transmitted to an output meter(s) and/or data storage means. Signal processing, for example, by a Digital Signal Processor or  $\square$ DSP, $\square$  can be employed before and/or after intensity data on the measured optical signals are transmitted to the output meter(s) and/or data storage means.

In flow cytometry, an "event" occurs when a cell or particle passes through the beam of the laser or other  
30       light source. As the event progresses, the light measured from scattering and/or fluorescence emission increases as the cell or particle enters the beam, reaches a maximum at the center of the beam, and tapers to a nominal value as it leaves the beam.

5 State of the art flow cytometers commonly use one of two systems to measure events. One system uses peak detectors, including peak hold circuits, to sustain the maximum signal level obtained from an event, and to measure optical parameters relating to cells and/or particles passing through a laser beam. Once a peak detector has sensed and measured an event, the peak detectors are turned off to provide sufficient time for an analog to digital, or [A/D,] converter to digitize that maximum value of the signal. During the time that the peak  
10 detectors are off, termed "dead time," any event occurring within the laser beam will go undetected.

The second system uses an integrator to measure the area under the peak collected for an event. With either system, when an event occurs within the cytometer's light beam during the dead time, the event will go undetected.

Normally, such lack of detection of an event does not constitute a significant problem because, for a  
15 large number of events, only a tiny number will be missed. However, applications which require the detection of rare, critical events, for example, identification of one special cell or particle in a thousand or a million, suffer from the possible non-detection of a rare event when current generation flow cytometers are used for detection and measurement. The higher the throughput, the greater is the probability that such a critical event will be missed per unit time period.

20 One solution to problems encountered in sorting samples in flow cytometry is provided by United States Patent No. 5,550,058 to Corio, et al., incorporated herein by reference. Corio, et al. provides a means of flexibly controlling decisions on the sorting of events detected in a flow cytometer. Events are pre-qualified according to user selectable parameters to permit reduction of events missed during dead time. However, the Corio, et al. reference does not teach or suggest a means of reducing the probability of missing a dead time  
25 event to zero.

Thus, it would be beneficial in the art of flow analyzers, including flow cytometers or other analyzing equipment, to have a means to reduce to zero the probability of missing any event, including a rare, dead time event.

It is also desirable to efficiently and inexpensively provide a system and/or method of reducing the  
30 probability of missing an event.

5           It is further desirable to efficiently and inexpensively provide a system and/or method of reducing the probability of missing an event for high sampling rates, for example, on the order of approximately a sampling period, such as one sample per millisecond or shorter period.

#### SUMMARY OF THE INVENTION

10           Accordingly, it is a feature and advantage of the present invention to provide an apparatus and a method for reducing the dead time to zero in flow analyzers, flow cytometers, and other measurement devices.

          It is another feature and advantage of the present invention to provide a means to reduce the dead time to zero in flow analyzers, flow cytometers, and other measurement devices by use of, for example, a circular buffer to store optical data until the data can be processed by at least one digital signal processor or other  
15   processor.

          It is a further feature and advantage of the present invention to provide a means to reduce the dead time to zero in flow analyzers, flow cytometers, and other measurement devices by providing software that permits the user to specify one or more of data collection, data storage, and data processing parameters, such as size of data storage areas, number of data storage areas, sampling rate (and/or the inverse: length of sampling or  
20   sample period), signal-to-noise (S/N) threshold, and fixed trailing distance.

          It is a feature and advantage of the present invention to efficiently and inexpensively provide a system and/or method of reducing the probability of missing an event.

          It is another feature and advantage of the present invention to efficiently and inexpensively provide a system and/or method of reducing the probability of missing an event in the presence of a high sampling rate.

25           It is yet another feature and advantage of the instant invention to provide the capability of enhancing data analytical capability by going back in time to re-examine and reprocess data in "old" data storage areas in order, potentially, to recreate a more complete event peak(s) by applying special algorithms to subthreshold data.

          The instant invention provides an apparatus and a method of use of that apparatus to reduce dead time to  
30   zero or substantially zero in a measurement device. The apparatus includes 1) a circular buffer that has a) a plurality of data storage areas that are linked to receive and consecutively store incoming data from a plurality of successive sampling periods of a measurement device that measures events occurring in a measurement zone,

- 5     b) a First Pointer (or first logical pointer), and c) a Second Pointer (or second logical pointer); and 2) at least one Digital Signal Processor (DSP) or other data processor.

Each data storage area is configured to receive and store data from one sampling period. Furthermore, the data storage areas are linked in an order that provides storage for data from a next-in-time sampling period into the next-in-order data storage area. The First Pointer directs receipt and storage of data of the next-in-time  
10     sampling period into the next-in-order data storage area; and, once all data storage areas contain data, the First Pointer directs receipt of data of the next-in-time sampling period, and storage by overwriting therewith the data in the next-in-order data storage area, which contains the oldest data of the circular buffer.

The Second Pointer is directed to the data storage area that stores data from a sample period preceding the current time by at least a fixed trailing distance, referenced to the First Pointer; and the Second Pointer  
15     directs one of the at least one DSPs to read and process data in the data storage area to which the Second Pointer is directed.

Once the next-in-time data from the current sampling period have been stored, the First Pointer advances to the next-in-order data storage area. In addition, once data in the data storage area to which the Second Pointer is directed have been processed, the Second Pointer advances to the next-in-order data storage area to  
20     repeat the cycle of data processing, followed by advancement to the next-in-order data storage area.

Other addressable buffers can alternatively be used where the addressable range of data allows the system to go forward and sufficiently backward to obtain previously collected samples. For example, a FIFO and backward cache combination can alternatively be used in the present invention, as can a cascading buffer.

The method of the instant invention entails applying the above apparatus to collect and process data from  
25     events occurring in a measurement device, particularly a measurement device with a flowcell, such as a flow cytometer or other flow analyzer.

The fixed trailing distance is set by the user to equal the longest possible time for an event, or longer, often falling in the 10 to 100 millisecond range. The size and number of data storage areas in the circular buffer also are user specified. Data storage areas often are set to 12 to 16 bits, though larger and smaller sizes may be  
30     used to adjust for particular applications. The number of data storage areas often is set to about one thousand, or greater. However, numbers of data storage areas less than a thousand also can be used advantageously.

The sampling rate, and/or its inverse which is length of sampling period, also are user specified. A

5 typical sampling rate is one million per second, which equates to sampling periods of one millisecond each. However, larger and smaller figures can be utilized according to the particular application.

When a plurality of DSPs is used, the user may specify the order of use of the respective DSPs. For example, one DSP can control storage of data incoming to the circular buffer, and one or a plurality of DSPs can handle processing of the data, as directed by the Second Pointer.

10 Buffers that are functionally equivalent to circular buffers also can be used, for example, a First In-First Out buffer which temporarily stores data (or lack thereof) from the respective data storage areas into one or more caches for possible later processing; double or triple cascading buffers; etc.

In another embodiment, an FPGA (Field Programmable Gate Array) may be programmed as a state machine, for example, with four or five states, to replace use of a DSP for inputting data into the circular buffer  
15 or functional equivalent.

The above broad outline of the more important features of the instant invention has been presented 1) to facilitate understanding of the following sections, which provide greater detail with respect to components of the instant invention, and 2) to succinctly highlight contributions of the instant invention to the art. However, it is to be understood that the above outline, as well as the terminology and details as to construction, arrangement  
20 and practice of the instant invention presented in the following sections, are exemplary and not limiting; i.e., the instant invention is capable of other embodiments as to construction, arrangement and practice.

As such, those skilled in the art will appreciate that the conception, upon which this disclosure is based, can readily be utilized as a basis for designing other structures and methods of practice to execute the purposes of the instant invention. Therefore, it is important that the instant disclosure and claims be regarded as  
25 embracing equivalent constructions and practices insofar as they do not depart from the spirit and scope of the invention as disclosed herein.

Another purpose for this SUMMARY OF THE INVENTION section is to enable the U.S. Patent Office and the public generally, including scientists, engineers and practitioner in the art who are not familiar with patent or legal terms or phraseology herein, to quickly determine the nature and essence of the instant disclosure  
30 with only cursory review. Thus, the intention of this section is not to limit the scope of the instant invention in any way.

For a more complete understanding of the instant invention, including its operating advantages and



5 various uses, reference should be made to the following drawings, descriptive matter, and claims, which illustrate preferred embodiments of the invention.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

Fig. 1 is an embodiment of a circular buffer according to the instant invention.

10 Fig. 2 is a flow chart of an embodiment for writing data from a flow analyzer to the circular buffer of Fig. 1.

Fig. 3 is a flow chart of an embodiment for reading and processing data from the circular buffer of Fig. 1.

Fig. 4 is a schematic showing the functioning of a circular buffer within a flow analyzer.

#### **DESCRIPTION OF THE PREFERRED EMBODIMENTS**

15 The present invention provides a system and method of reducing the dead time to zero or substantially zero in flow analyzers, flow cytometers, and other measurement devices. Advantageously, the present invention provides the user the ability to specify one or more data collection, data storage, and data processing parameters, such as size of data storage areas, number of data storage areas, sampling rate (and/or the inverse: length of sampling or sample period), signal-to-noise (S/N) threshold, and fixed trailing distance.

Each data storage area is configured to receive and store data from, for example, one or more sampling periods. The data storage areas are optionally linked in an order that provides storage for data from a next-in-time sampling period into the next-in-order data storage area. A First Pointer (or first logical pointer) directs receipt and storage of data of the next-in-time sampling period into the next-in-order data storage area. Once all  
25 data storage areas contain data, the First Pointer directs receipt of data of the next-in-time sampling period, and storage by overwriting therewith the data in the next-in-order data storage area, which contain the oldest data of the buffer, such as a standard circular buffer.

A Second Pointer (or second logical pointer) is directed to the data storage area that stores data from a sample period preceding the current time by at least a fixed or predetermined trailing distance with reference, for example, to the First Pointer or other reference. The Second Pointer directs at least one digital signal  
30 processor (DSP) or other data processor to read and process data in the data storage area to which the Second Pointer is directed.

5        Once the next-in-time data from the current sampling period have been stored, the First Pointer advances to the next-in-order data storage area. In addition, once data in the data storage area to which the Second Pointer is directed have been processed, the Second Pointer advances to the next-in-order data storage area to repeat the cycle of data processing, followed by advancement to the next-in-order data storage area.

10        The present invention is based, in part, on my discovery that samples containing important data were being lost in data collection of analyzing equipment, such as flow analyzers or cytometers. As a result, the overall accuracy of the cytometer was questionable or could be improved. Accordingly, having realized that important samples were being lost for certain applications described above, I determined that an addressable buffer having the capability of obtaining a predetermined amount of data backward in time would improve the overall accuracy and efficiency of the flow analyzer or cytometer.

15        Other addressable buffers can alternatively be used where the addressable range of data allows the system to go forward and sufficiently backward to obtain previously collected samples. For example, a FIFO and backward cache combination can alternatively be used in the present invention, as can a cascading buffer. The number of samples for which the buffer need go backward in time varies, depending on the application, but may be anywhere from 1 sample, 20 samples, and higher.

20        Referring to Fig. 1, an embodiment of a circular buffer (also referred to as a "circular buffer means" or "circular memory structure") 100 having 30 data storage areas is presented. The 30 data storage areas are represented by consecutive even numbers 102 through 160. First Pointer (or first logical pointer) 170 indicates the data storage area to which the next data sample from the next sample period is to be stored by, for example, overwriting the existing data (if any) when the next data sample is transmitted from a flow analyzer, flow  
25        cytometer, or other measurement device to circular buffer 100.

30        First Pointer 170 is directed to data storage area 102, which, at this particular instant in time, holds the oldest data of the circular buffer, as is indicated by a single asterisk. When data are transmitted to circular storage buffer at the beginning of an analytical procedure, instead of overwriting old data, new data are added consecutively to the data storage areas of circular buffer 100, thus creating a series of linked next-in-time data storage areas.

      Circular arrow 190 indicates the forward-in-time direction of advancement both for data acquisition and usual data processing (i.e., not back-in-time processing). Thus, data storage area 104 holds the second oldest

5 data at this instant in time, 106 holds the third oldest data, etc., to data storage area 160, which holds the newest data, as is indicated by a double asterisk. The next set of incoming data from a sample period will overwrite the old data of 102.

Just prior to arrival of the next set of incoming data transmitted to circular buffer 100 from the next sample period, First Pointer 170 advances to data storage area 104, to direct overwriting of the (now) oldest data  
10 consecutively for each data storage area, thus continuously advancing in the forward-in-time direction of circular arrow 190. This process continuously creates and updates a series of next-in-order data storage areas that store next-in-time data. Referring again to Fig. 1, Second Pointer (or second logical pointer) 180 is shown as being directed to data storage area 156 of circular buffer 100. Second Pointer 180 indicates the current data storage area whose data are to be read next and processed by, for example, a DSP or other  
15 processor (not shown in the figure) if the data reach a user specified threshold signal-to-noise (S/N) ratio. After the DSP reads the data stored in data storage area 156, Second Pointer 180 advances in the direction of arrow 190 to data storage area 158, and so forth in sequence. As indicated above, other buffer devices may be used that are addressable; therefore, there is no requirement that the above operations necessarily be performed sequentially or in a specific linear manner or order.

20 The number of data storage areas in a circular buffer or other buffer is flexibly specified by the user to be as large or as small as the requirements of the particular application. For example, a thousand data storage areas provide sufficient means to store one thousand sampling periods, where each sampling period is one microsecond. Circular buffer 100 can comprise more than or less than 1,000 data storage areas; and the length of time of a sampling period can be greater than or less than one millisecond (i.e., the sampling rate can be less  
25 than or greater than one million per second).

Similarly, the size of the data storage areas is defined by the user or is user selectable in concert with the requirements of the particular application. For example, a capacity of 12-16 bits per data storage area will suffice for some applications, whereas more than 20 bits will be required in other applications, such as complex and/or high amplitude fluorescence emission or other peaks.

30 Practice of the present invention does not require a one-to-one (i.e., in locked-step) direct correspondence between the quantity of data read and processed by the DSP (or the time required therefor), as directed by Second Pointer 180, and the quantity of incoming data that overwrites the oldest data (or the time required

5 therefor), as directed by First Pointer 170. For example, First Pointer 170 advances in steady cadence in step with the sampling rate, for example, one million per second. When data in a data storage area do not meet the user specified S/N threshold, Second Pointer 180 also moves in cadence with the sampling rate.

On the other hand, when data in a data storage area do meet a user specified S/N threshold and thus indicate an event in progress, Second Pointer 180 advances quickly to the next-in-time data of the next-in-order  
10 data storage area. As long as the data in a data storage area do not meet the user specified S/N threshold, a minimum, fixed lag time  $\tau$  or fixed trailing distance is maintained relative to First Pointer 170, which lag time must be at least equal to  $\tau_{mc}$ , the longest time for an event. Thus, the fixed trailing distance serves as a minimum lag time criterion when data in a data storage area do not meet the S/N threshold.

In other words,  $\tau$  is the user defined lag time between the current transmission of data from a sampling  
15 period to circular buffer 100, and the reading by the DSP of insufficient S/N data stored  $\tau$  sec. previously. Thus, the DSP is programmed to examine data collected from a point back in time  $\tau$ , which is equal to or greater than  $\tau_{mc}$ , measured from the time that First Pointer 170 is directed to the data storage area into which the next set of incoming data (from the next sampling period) are to be stored.

Stated differently, when First Pointer 170 advances to the next-in-order data storage area, Second Pointer  
20 180 also advances to the next data storage area if data in the preceding data storage area did not meet the S/N threshold. However, if those data did reach the S/N ratio, then Second Pointer 180 advances rapidly ahead to permit the DSP to read and process the data of that storage area forward in time, as well as to read and processes the data of subsequent next-in-order data storage areas relating to the event. Thus, when an event is detected, Second Pointer 180 advances in the direction of circular arrow 190 to the next-in-order data storage area with  
25 the next-in-time data for the event. In so doing, Second Pointer 180 will "overrun" fixed lag time  $\tau$ , but will not overrun First Pointer 170 because  $\tau$  is at least equal to the longest possible event.

Alternative embodiments can employ a plurality of DSPs or other standard processors to increase processing efficiency as Second Pointer 180 advances rapidly to process an event. With such a plurality of DSPs, the user can specify their order of processing as Second Pointer 180 advances through the series of data  
30 storage areas with data from the event.

In addition, more sophisticated standard protocols or other standard protocols can be employed for determining whether data in a given data storage area belong to a peak from an event. For example, when each

5 set of data in a series of data storage areas does not meet the user specified S/N threshold, but the values of each set of data from the series of data storage areas are monotonically increasing or otherwise substantially increasing by a user specified amount, and/or within a few data storage areas the increasing sets of data mature into a peak or other recognizable event, then Second Pointer 180 retreats to the first-in-order data storage area with data now deemed (by virtue of the logic programmed into the software therefor) consistent with the  
10 occurrence of an event. Therefore, the data of such a series of data storage areas are thereby processed as being part of a peak of an event.

This is the reason for the requirement of having a circular buffer or a functional equivalent; i.e., a means to reexamine previous data in view of data subsequent in time that confirm the presence of an event. Thus, this embodiment provides powerful additional analytical power over prior art flow analyzers and other measurement  
15 devices. The term circular buffer (and other equivalent expressions, such as "circular buffer means" and "circular memory structure") refers to a memory structure or device comprising data storage areas that are linked or related in a prescribed order in a fashion that permits a number of functional capabilities. One functional capability comprises the capability of ordered or predetermined filling (for example, next-in-time or other addressable fashion) of those data storage areas with consecutive packets of data. Alternatively, or in  
20 addition, another functional capability is to store data and permit the data to be output in a predetermined manner, while also allowing a predetermined amount of back or past data to be retrieved therefrom.

Once all of the discrete number of data storage areas each have a packet of data, a second cycle of ordered filling commences with the next-in-time data overwriting the data in the data storage area having the oldest data, and so forth for all subsequent cycles, where each cycle transmits consecutive data packets into data  
25 storage areas in the same order or in the same addressable fashion as the previous cycle.

A second functional capability is that of permitting a pointer (for example, Second Pointer 180) to go back in time and reexamine data in view of subsequent data, per one or more user specified criteria. In addition, it is important to recognize that other buffer structures (whether presently available or developed in the future) that accomplish results that are functionally equivalent to those described herein above, as required by the instant  
30 invention, are encompassed by the instant invention.

Two non-limiting examples can be mentioned: 1) cascading double or triple buffers, and/or 2) First In-First Out buffers which are coupled with at least one cache to which data from each respective data storage area

5 are transferred when Second Pointer 180 advances to the next data storage area. In addition, any other buffer exhibiting the above two functional capabilities are hereby explicitly encompassed by the instant invention.

Referring to Fig. 2, the processing logic of data transfer to circular buffer 100 of Fig. 1 is shown. Step 200 shows that First Pointer 170 is directed to the storage area of circular buffer 100 having the oldest data. Sample Period N then occurs 202, and sensors sample and measure light parameters, and high speed analog to digital  
10 (A/D) converters digitize the data 204. Digitized data from the sensors are then transmitted 206 to the data storage area to which First Pointer 170 is directed, and the oldest data are overwritten. When the length of a sample period (inverse of the sampling frequency) has passed, measured from the position of First Pointer 170 at step 200, First Pointer 170 advances 210 to the next-in-order data storage area or predetermined data storage area, which has the next-in-time data, which now are the oldest data in the circular buffer; and the cycle  
15 (equivalent to the sequential combination of steps 204, 206, and 210) continues, beginning with the next sampling period N + 1 (212).

In the case of an analytical process that is just starting, the equivalent of steps 200 - 206 will apply, though the data storage areas initially will be empty, because there are no data to be overwritten. For example, if the sampling rate is one million per second, and if there are one thousand data storage areas, such a state of empty  
20 data storage areas will exist for substantially only one thousandth of a second.

For purposes of analysis, an event is defined by the user with respect to a minimum datum criterion, for example, when the signal-to-noise (S/N) ratio is equal to a threshold value of about 2 or 3, or other threshold value. The S/N threshold value may be increased or decreased to compensate for the degree of purity of the samples (for example, whether integrity of the particles is high, or many of the particles have crumbled into  
25 smaller entities) and/or the purity of the fluid, or to compensate for any other situation (for example, bubbles) affecting background noise in the cytometric or other flow measurement readings, or in readings from any other measurement device.

In effect, the S/N threshold may be adjusted to reflect the range of anticipated small signal amplitudes, particularly including amplitudes of background signals. As is indicated above in the discussion of Fig. 1, more  
30 sophisticated S/N threshold strategies can be employed when the application will benefit.

The rate of sampling also is selected by the user, for example, one million sampling periods per second, or smaller or greater. Typically, a fixed rate of oversampling is used that creates about 8-12 sampling periods per

5 average event, with a maximum of about 20 sampling moments per event. However, greater or lesser values may be used according to a particular application; for example, when there is a complex or broad, multi-peak fluorescence emission spectrum, more than 50 or 100 sampling periods per event can be specified advantageously.

Referring now to Fig. 3, at times not distant from the measuring, transmitting, and storing of new data, as  
10 outlined by consecutive steps 200, 202, 204, 206, and 210 of Fig. 2, data processing occurs with respect to data stored elsewhere in circular buffer 100, starting with step 300. At step 300, Second Pointer 180 is directed to the data storage area that stores data collected no less than  $\tau$  sec. previously, which period of time is also referred to as the "fixed trailing distance." The DSP then reads the data 302, and determines 304 whether the data reach the user selected S/N threshold so as to warrant processing 306 by the DSP.

15 If the S/N ratio threshold has been reached, the data in that data storage area are processed 306, Second Pointer 180 advances to the next-in-order data storage area 330, and a determination is made as to whether the data in that storage area reach the S/N threshold 304.

On the other hand, if the S/N ratio threshold has not been reached by the data in the particular data storage area, then more sophisticated standard analytical strategies are employed, such as asking if there is an upward  
20 trend in consecutive subthreshold data 340. If the answer is "yes," then Second Pointer 180 goes back to data at the beginning of the trend 350, and processes the data in the series of consecutive data storage areas that constitute the "trend" 360. Subsequently, Second Pointer 180 advances to the data storage area with the next-in-time data 370, and the cycle begins again with the question "Has the S/N threshold been reached" 304.

Alternative or additional algorithms can be employed at steps 340 and/or 360. For example, a "small signal  
25 detector algorithm" that facilitates inter-processor (for example, inter-DSP) communication with respect to data that do not meet the S/N threshold. For example, when a plurality of DSPs is employed, such an algorithm will direct each of the plurality of DSPs to cooperatively go back in time to preceding data that, individually or collectively, may constitute a portion of an event peak that is developing. Likewise, such an algorithm also can be employed for the "end" portion of an event peak. In effect, this kind of algorithm signals the need for  
30 additional scrutiny by one or more analytical programs that can include, for example, fitting and/or smoothing and/or rounding and/or other programs that provide a basis for additional signal processing.

On the other hand, if there is no upward trend in subthreshold data (or if any other user specified criterion

5 applies, such as length of an event), then a pause or wait 308 is made. Next, the question "Has lag time  $\tau$  been met?" is posed 310; i.e., were the data in the data storage area collected and stored no less than  $\tau$  sec. previous to current time, as measured relative to First Pointer 170? If the answer is "No," a further wait 308 is imposed, and cycling back to the question of step 310 is performed. If the answer is "Yes," Second Pointer 180 then advances 320 to the next-in-order data storage area that has the next-in-time data. The particular data storage  
10 area to be examined next is always the one with data stored from the sampling period immediately following the sampling period just examined, i.e., next-in-time (see direction of circular arrow 190 in Fig. 1). Subsequently, a new cycle of data reading and processing commences.

A number of such cycles of collecting and processing of data can occur before the next event occurs. For example, if there are one million sampling periods per second (i.e., the sampling rate), and if 20,000 cells and/or  
15 particles ("events") are read per second, which is the fastest rate currently commercially available, then the average eventless time between samples is about 40 milliseconds if 10 sampling periods are required to collect the data of a single event. When circular buffer 100 is selected to have one thousand data storage areas, then data in those thousand data storage areas represent sampling over a period of one thousandth of a second.

Thus, the average eventless time of about 40 milliseconds translates into an average of about 40 sampling  
20 periods with no detectable data until the next event. Calculation also shows that, at a given instant in time, a circular buffer with 1,000 data storage areas will have data from about 20 events [ $1,000 \text{ milliseconds} / (10 \text{ milliseconds event length} + 40 \text{ milliseconds eventless time})$ ]. Data storage areas holding data for inter-event periods are filled with background noise data.

In a preferred embodiment, simultaneous data samples (for example, from a single sampling period) from  
25 each of a plurality of channels of data that have been detected and digitized are stored in a single data storage area. For example, a detector for one channel will collect and transmit data on scattered excitation light; a detector for a second channel will collect and transmit data on fluorescence emission data of a first fluorescent dye; a detector from a third channel will collect and transmit data on fluorescence emission from a second fluorescent dye; and all of these data are stored in or on a single data storage area.

30 A larger number of data storage areas may be employed for simultaneous data from a multitude of channels, with order of processing being specified by the user. For example, three channels can be used to collect emission data from three spectrally distinct fluorescent dyes; one channel can collect forward scattered



5 light; one channel can collect side scattered light; and the order for processing data from the five channels is specified by the user. Alternatively, five separate data storage areas can be used respectively to store data from the five channels, where the order of storage is user specified.

In another embodiment, more than one DSP can be employed to work in concert with Second Pointer 180 as a solution to the problem that processing activities often take considerable time. For example, time consuming digital filtering (for example, standard finite impulse response filtering, or "FIR") can be used, and/or any of various waveform analysis procedures that are often employed to improve the quality of data extracted in the presence of noise. When more than one DSP is so employed, the user specifies the order of processing by the respective DSPs or other processors. For example, a first DSP processes data of the next-in-order data storage area, a second DSP processes data of the (now) next-in-order data storage area, and so forth. Other configurations are also possible in accordance with the present invention.

It is beneficial that the present invention permits the user to specify not only the S/N threshold, but also optionally the order of transmission and processing of multiple channels of data from events, the order of processing when there are multiple DSPs, the size of a data storage area, the number of data storage areas and/or the total size of circular buffer. Thus, in instances of complex emission peaks, multiple data inputs, or any other complicated data stream, the user can readily adapt the data storage parameters and other parameters to fit the current use, thus ensuring efficient processing and ample data storage capacity to accommodate at least the largest possible quantity of data resulting from one or more events occurring in close time proximity.

In a preferred embodiment, circular buffer 100 has 1,000 or more data storage areas that each hold all incoming data from two or more channels (or three or more in complex analytical applications). A typical minimum configuration includes one channel collecting scattered light from excitation light, a second channel collecting fluorescence emission light from a first fluorescent dye, and a third channel collecting fluorescence emission light from a second fluorescent dye. However, depending on the application, as few as one channel or as many as five or more channels can provide the incoming data to circular buffer 100. The upper limit of channels is immaterial to the present invention, but more than five channels clearly is within the scope of the present invention.

Referring to Fig. 4, a schematic is shown for functioning of circular buffer 100 within a flow analyzer 420, for example, a flow cytometer. Laser light beam 402 from laser 400 is shown to impinge on a particle 406 as it

5 passes through examination zone 404 of a flow cuvette. Light 422, 424 that is scattered, emitted (for example, fluorescence emission), refracted, or otherwise produced is then sensed by one or more detectors 408, 410, which then transmit their data 426, 428 to their respective FADC (filtering, amplification, and digital conversion) units 412, 414. The FADCs then transmit 430, 432 their respective resulting digitized data to circular buffer 100. By the processes described herein above, circular buffer 100 then transmits 434 data  
10 meeting the user specified S/N threshold to a DSP or other data processor 416 for final data processing. Results of processing by the DSP or other processor 416 are then output 436 as the final, processed data 418. It is important to recognize that circular buffer 100 may comprise a part of DSP or other processor 416, or be operatively coupled thereto.

Also, it is important to recognize that other buffers can be used in place of circular buffer 100, and yet  
15 fulfill the accomplishments described herein above with respect to circular buffer 100, for example, cascading buffers, and First In-First Out buffers which utilize a cache for re-reading and/or reprocessing data when so directed by Second Pointer 180.

The present invention is directed to use with flow cytometers, flow analyzers, other flow measurement devices, etc. The processing logic of the present invention that, in effect, permits digitized signals to be stored  
20 rapidly, and thereafter be processed by at least one DSP or other processor (using eventless time periods to conduct the time consuming procedures of signal processing), can be applied equally well to other measurement applications.

Attached hereto as part of the instant Specification in the APPENDIX is a copy of an assembly language program that implements one embodiment of the present invention.

## 5 WHAT IS CLAIMED IS:

1. A data collection and measurement device collecting data with substantially zero dead time,  
comprising:

a) a circular buffer comprising:

10 a plurality of data storage areas that are linked to receive and consecutively store incoming data from  
a plurality of successive sampling periods for at least one event of a measurement device, where each data  
storage area is configured to receive and store data from one sampling period, and the data storage areas  
are linked in an order that provides storage for data from a next-in-time sampling period into the next-in-  
order data storage area;

a First Pointer; and

15 a Second Pointer; and

b) at least one Digital Signal Processor (DSP), where the First Pointer directs receipt and storage of data  
of the next-in-time sampling period into the next-in-order data storage area, and, once all data storage areas  
contain data, directs receipt of data of the next-in-time sampling period, and storage by overwriting therewith  
20 data in the next-in-order data storage area, the Second Pointer is directed to the data storage area that stores data  
from a sample period preceding a current time of data receipt and storage by a fixed trailing distance as  
referenced to the First Pointer, and directs one of the at least one DSPs to read and process data in the data  
storage area to which the Second Pointer is directed, and wherein once the next-in-time data have been stored,  
the First Pointer advances to the next-in-order data storage area, and once data in the data storage area to which  
25 the Second Pointer is directed have been processed, the Second Pointer advances to the next-in-order data  
storage area, and

wherein said collection and measurement apparatus collects and processes data from the at least one event.

2. The data collection and measurement device of claim 1, where the measurement device is a flow  
30 cytometer.

3. The data collection and measurement device of claim 1, where the fixed trailing distance equals at least

5 a longest possible time for an event.

4. The data collection and measurement device of claim 3, where the fixed trailing distance is about 10 to about 100 milliseconds.

10 5. The data collection and measurement device of claim 1, where the fixed trailing distance is user specified.

6. The data collection and measurement device claim 1, where the sampling periods are of a fixed length of time, and the fixed length of time is user specified.

15

7. The data collection and measurement device of claim 6, where the sampling periods are each about one millionth of a second.

20

8. The data collection and measurement device of claim 6, where the sampling periods are each less than one millionth of a second.

9. The data collection and measurement device of claim 1, where the number of data storage areas is user specified.

25

10. The data collection and measurement device of claim 9, where the number of data storage areas is at least one thousand.

11. The data collection and measurement device of claim 1, where each data storage area has a size, and the size is user specified.

30

12. The data collection and measurement device of claim 11, where the size of each data storage area is from about 12 bits to about 16 bits.

5

13. The data collection and measurement device of claim 11, where the size of each data storage area is greater than 16 bits.

10

14. The data collection and measurement device of claim 11, where the size of each data storage area is less than 12 bits.

15. The data collection and measurement device of claim 1, further comprising an order of data processing by DSPs when a plurality of DSPs is employed.

15

16. The data collection and measurement device of claim 15, where the order of data processing by DSPs is user specified.

20

17. The data collection and measurement device of claim 1, further comprising a signal-to-noise ratio threshold that serves as a criterion as to whether data are processed by the at least one DSP.

18. The data collection and measurement device of claim 17, where the signal-to-noise ratio threshold is user specified.

25

19. The data collection and measurement device of claim 17, further comprising logic to re-examine and reprocess previous data that do not meet the signal-to-noise ratio threshold.

30

20. A flow analyzer collecting data with substantially zero dead time for at least one event, comprising:  
at least one circular buffer having a plurality of data storage areas that are adapted to receive and store incoming data from a plurality of sampling periods for the at least one event collected by said flow analyzer, said at least one circular buffer including at least a First Pointer used to direct receipt and storage of the data

- 5 collected by said flow analyzer into at least one of said plurality of data storage areas including overwriting of stored data therein in accordance with predetermined criteria, and at least a Second Pointer used to direct reading of the data from at least one of the plurality of data storage areas including previously stored data from a sample period preceding a current time of the receipt and the storage of the data directed by the at least First Pointer; and
- 10 at least one processor connected to said at least one circular buffer, receiving the data from the at least one of the plurality of data storage areas of said at least one circular buffer responsive to direction by the at least one Second Pointer, and said flow analyzer collecting and processing the data from the at least one event with substantially zero dead time.

15 21. The flow analyzer of claim 20, where the flow analyzer is a flow cytometer.

22. A flow analyzer collecting data with substantially zero dead time for at least one event, comprising:  
at least one buffer having a plurality of data storage areas that are adapted to receive and store incoming data from a plurality of sampling periods for the at least one event collected by said flow analyzer, said at least  
20 one buffer including previously stored data from a sample period preceding a current time of the receipt and the storage of the data and including an addressable range of the plurality of data storage areas to go sufficiently backward in the plurality of sampling periods;

at least one processor connected to said at least one buffer, receiving the data from the at least one of the plurality of data storage areas of said at least one buffer, and said flow analyzer collecting and processing the  
25 data from the at least one event with substantially zero dead time by reading from said at least one buffer the previously stored data from the sampling period preceding the current time of the plurality of sampling periods.

23. The flow analyzer of claim 22, where the flow analyzer is a flow cytometer.

30 24. A flow analyzer collecting data with substantially zero dead time for at least one event, comprising:  
at least one buffer means for having a plurality of data storage areas that are adapted to receive and store incoming data from a plurality of sampling periods for the at least one event collected by said flow analyzer,

5 said at least one buffer means for including previously stored data from a sample period preceding a current time of the receipt and the storage of the data and including an addressable range of the plurality of data storage areas to go sufficiently backward in the plurality of sampling periods;

at least one processor for receiving the data from the at least one of the plurality of data storage areas of said at least one buffer means, and said flow analyzer collecting and processing the data from the at least one  
10 event with substantially zero dead time by reading from said at least one buffer means the previously stored data from the sampling period preceding the current time of the plurality of sampling periods.

25. The flow analyzer of claim 24, where the flow analyzer is a flow cytometer.

15 26. A method of collecting data used in a flow analyzer with substantially zero dead time for at least one event, comprising the steps of:

(a) receiving incoming data from a plurality of sampling periods for the at least one event collected by said flow analyzer via at least one circular buffer having a plurality of data storage areas;

(b) storing the incoming data in the at least one circular buffer via at least a First Pointer used to direct the  
20 receipt and the storage of the data collected by said flow analyzer into at least one of the plurality of data storage areas including overwriting of stored data therein in accordance with predetermined criteria;

(c) directing the reading of the data by at least one processor from at least one of the plurality of data storage areas including previously stored data from a sample period preceding a current time of the receipt and the storage of the data directed by the at least First Pointer via at least a Second Pointer;

25 (d) receiving the data by the at least one processor from the at least one of the plurality of data storage areas of said at least one circular buffer responsive to direction by the at least one Second Pointer; and

(e) collecting and processing the data from the at least one event with substantially zero dead time.

27. The flow analyzer of claim 26, where the flow  
30 analyzer is a flow cytometer.

**AMENDED CLAIMS**

[received by the International Bureau on 28 October 1999 (28.10.99);  
original claims 1 and 27 amended; remaining claims unchanged  
(2 pages)]

1. A data collection and measurement device collecting data with substantially zero dead time, comprising:

a) a flow analyzer;

5 b) a circular buffer comprising:

a plurality of data storage areas that are linked to receive and consecutively store incoming data from a plurality of successive sampling periods for at least one event of the flow analyzer, where each data storage area is configured to receive and store data from one sampling period, and the data storage areas are linked in an order that provides storage for data from a next-in-time sampling

10 period into the next-in-order data storage area;

a First Pointer; and

a Second Pointer;

c) at least one Digital Signal Processor (DSP), where the First Pointer directs receipt and storage of data of the next-in-time sampling period into the next-in-order data storage area, and, once  
15 all data storage areas contain data, directs receipt of data of the next-in-time sampling period, and storage by overwriting therewith data in the next-in-order data storage area, the Second Pointer is directed to the data storage area that stores data from a sample period preceding a current time of data receipt and storage by a fixed trailing distance as referenced to the First Pointer, and directs one of the at least one DSPs to read and process data in the data storage area to which the Second pointer  
20 is directed, and wherein once the next-in-time data have been stored, the First Pointer advances to the next-in-order data storage area, and once data in the data storage area to which the Second Pointer is directed have been processed, the Second Pointer advances to the next-in-order data storage area, and wherein said collection and measurement apparatus collects and processes data from the at least one event.

25 2. The data collection and measurement device of claim 1, where the flow analyzer is a flow cytometer.

3. The data collection and measurement device of claim 1, where the fixed trailing distance equals at least



said at least one buffer means for including previously stored data from a sample period preceding a current time of the receipt and the storage of the data and including an addressable range of the plurality of data storage areas to go sufficiently backward in the plurality of sampling periods;

5 at least one processor for receiving the data from the at least one of the plurality of data storage areas of said at least one buffer means, and said flow analyzer collecting and processing the data from the at least one event with substantially zero dead time by reading from said at least one buffer means the previously stored data from the sampling period preceding the current time of the plurality of sampling periods.

25. The flow analyzer of claim 24, where the flow analyzer is a flow cytometer.

10 26. A method of collecting data used in a flow analyzer with substantially zero dead time for at least one event, comprising the steps of:

(a) receiving incoming data from a plurality of sampling periods for the at least one event collected by said flow analyzer via at least one circular buffer having a plurality of data storage areas;

15 (b) storing the incoming data in the at least one circular buffer via at least a First Pointer used to direct the receipt and the storage of the data collected by said flow analyzer into at least one of the plurality of data storage areas including overwriting of stored data therein in accordance with predetermined criteria;

20 (c) directing the reading of the data by at least one processor from at least one of the plurality of data storage areas including previously stored data from a sample period preceding a current time of the receipt and the storage of the data directed by the at least First Pointer via at least a Second Pointer;

(d) receiving the data by the at least one processor from the at least one of the plurality of data storage areas of said at least one circular buffer responsive to direction by the at least one  
25 Second Pointer; and

(e) collecting and processing the data from the at least one event with substantially zero dead time.

27. The method of claim 26, wherein the flow analyzer is a flow cytometer.



2/4

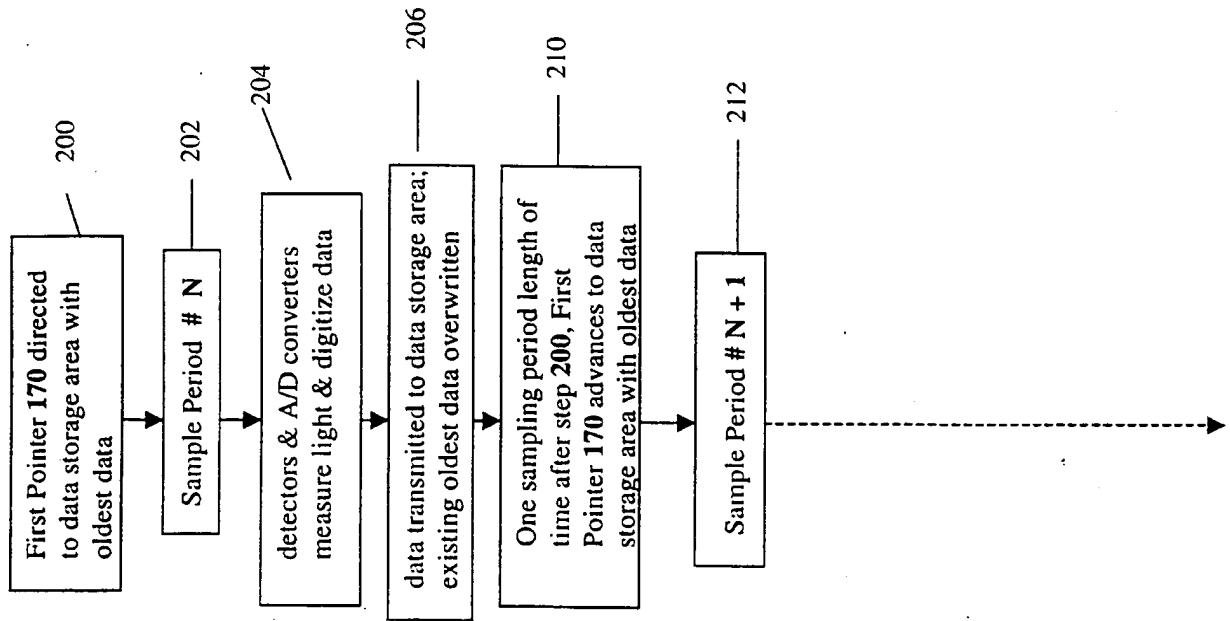


FIG. 2

Fig. 3

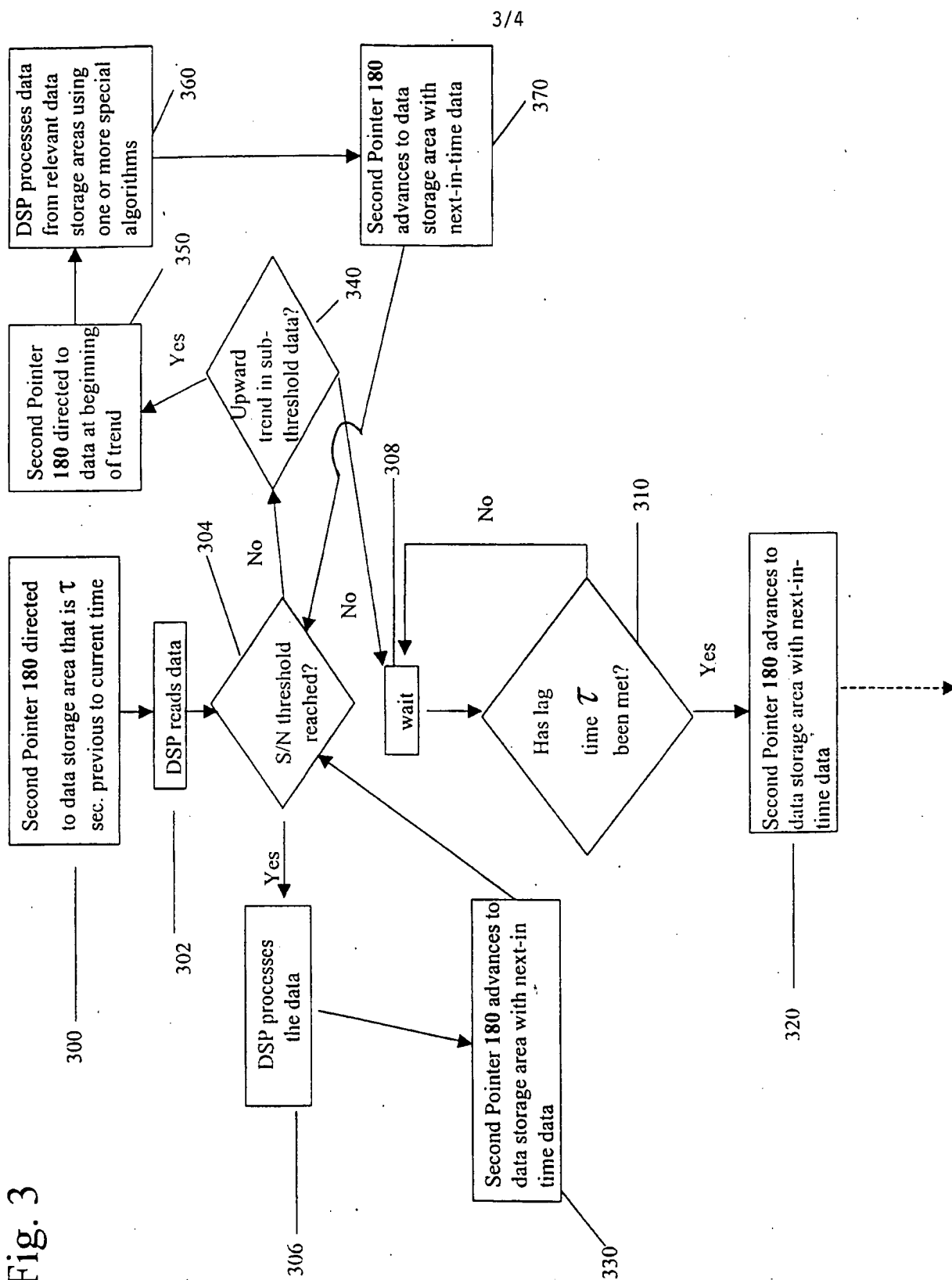
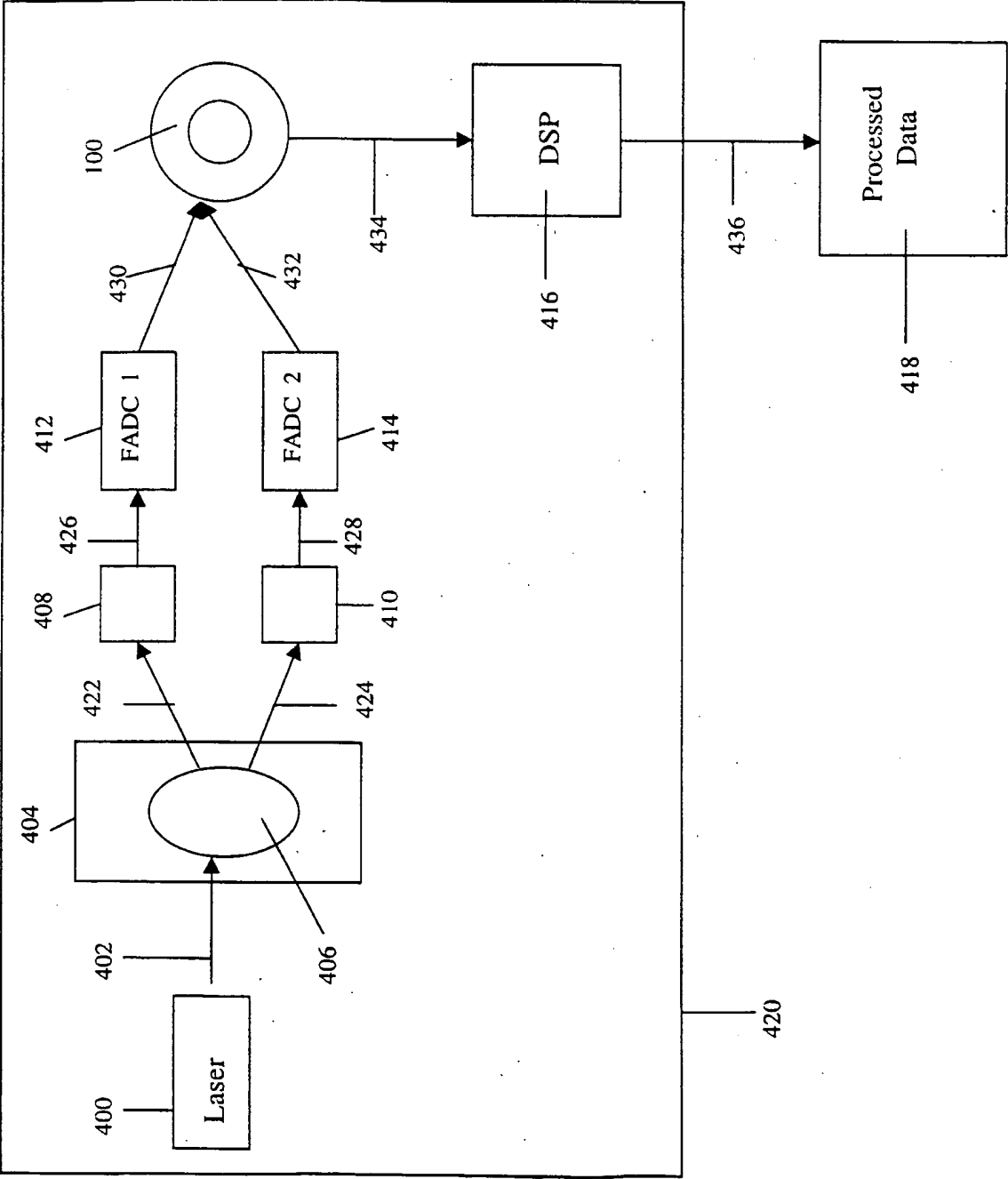


FIG. 4



.MODULE/ABS=0 LX1;

.pagelength 44;

/\* This software is written in the native assembly language  
of the Analog Devices 2181 Digital Signal Processor.

The purpose of this code is to oversample bead fluorescence  
in each of 5 consecutive channels to isolate single bead events  
and report the fluor area in each channel. The program  
double buffers the data for the 167 micro, and uses the  
following interrupts for interprocessor signalling

From 167 to IRQ this program:

p2.0\_\_IRQ0 go to idle state

p2.1\_\_IRQ1 new data in control area

p2.2\_\_FI begin capture (this is a state, not IRQ)

p2.3\_\_PWRDN nmi begin execution at PM 0x2c

From this program to IRQ 167:

FL0\_\_EX0IN buffer 0 is ready

FL1\_\_EX1IN buffer 1 is ready

FL2\_\_EX2IN error occurred during capture state

All PM and DM are accessible from the 167 thru IDMA so  
mem areas are assigned and restricted.

PM is loaded from EPROM but may be rewritten by:

1) IRQ0 (p2.0)

2) rewrite with vector to first instruction at 0x2c

3) PWRDN nmi (p2.3)

DM allocation (in hex) is:

0-ff reserved for 2181

100-1ff DSP status area, read only to 167

200-2ff DSP control area, read all/ write restricted to 167

300-3ff DSP command area, read/write 167, read / restricted write to DSP

400-9ff buffer 0, 1536 words, 384 max block size

a00-fff buffer 1

1000-3dff reserved for DSP raw A/D data

\*/

.CONST PFLAGS=0X3FE5;

/\* DSPs variable area \*/

.VAR/dm/ABS = 0 version, actbuf, pki5, wid, svi6, svavg, svlft, fstart, lls, hls, svi3, svi7, svcry, haf, lo5, recbuf[5], left;

.VAR/dm/ABS = 0x80 bkgnd[10];

.VAR/dm/ABS = 0x90 inbkgnd[10];

.VAR/dm/ABS = 0xa0 x8bkgnd[10];

.VAR/dm/ABS = 0xb0 histbk[10];

/\* DSP status \*/

.VAR/dm/ABS = 0X100 misscnt, xx1, buf0cnt, xx2, buf1cnt, xx3, otrcnt, xx4, afreg, xx5, nobgsub;

/\* DSP control \*/

.VAR/dm/ABS = 0X200 buf0siz, xa1, buf1siz, xa2, thresh, xa3, mcready, xa4, clock, xa5,

lena, xa6, bcntl, xa7, bcnth;

/\* DSP command \*/

.VAR/dm/ABS = 0X300 address, xxb1, hiword, xxb2, lobyte, xxb3, msg;

.VAR/dm/ABS = 0X400 outbuf0;

.VAR/dm/ABS = 0Xa00 outbuf1;

/\* raw data area \*/

.VAR/dm/abs = 0x1000 hist[0x1000] ;

.VAR/dm/circ/abs = 0x2000 workbuf[0x1e00] ;

```
/* interrupt vectors */

dis ints;
jump begin;      /*reset irq*/
nop;
nop;

jump irqfl;      /*read 4 fluorescence measurements*/
nop;
nop;
nop;

rti;             /*irqL1 */
nop;
nop;
nop;

rti;             /*irqL0 */
nop;
nop;
nop;

rti;             /*sp0 tx*/
nop;
nop;
nop;

rti;             /*sp0 rx*/
nop;
nop;
nop;

rti;             /*irq E */
nop;
nop;
nop;

jump bdmadone;   /*irq BdmA*/
nop;
nop;
nop;

jump procnewctl; /*irq1-new data in control area*/
nop;
nop;
nop;

jump doidle;     /*go to idle mode, wait for nmi*/
nop;
nop;
nop;

jump tim_exp;    /*timer*/
nop;
nop;
nop;

softload:
m0=0;
i0=0;
i0=^address;
```





```

dm(lena)=ax0;
m0=2;
m1=-1;
m2=1;
m3=0;
m4=-20;
m5=1;
m6=5;
m7=0;
i1=0x3fff;
dm(i1,m1)=0;
dm(i1,m1)=0x3ff8;      /* atod is ZERO wait state */
dm(i1,m1)=0;      /* tperiod */
dm(i1,m1)=0;      /* tcount */
dm(i1,m1)=15;      /* tscale to 1 us per cnt */
dm(i1,m1)=0;      /* sp0 */
dm(i1,m1)=0;
dm(i1,m1)=0;
dm(i1,m1)=0;
dm(i1,m1)=0;
dm(i1,m1)=0;      /* 3ff6 */
dm(i1,m1)=0;
dm(i1,m1)=0;
dm(i1,m1)=0;
dm(i1,m1)=0x4000;      /* ena internal clock gen */
dm(i1,m1)=15;      /* ser clock out to 1.024 MHz */
dm(i1,m1)=0;      /* addr 3ff0 */
dm(i1,m1)=0xd000;      /* set up power down options */
i1 = 0x3fe5;
dm(i1,m2)=0x0;      /* set pf0-7 to 0 */
dm(i1,m1)=0x78ff;      /* pf0-7 are outputs */
reset f10;
reset f11;
reset f12;
reset flag_out;      /* set flags for reg control */
ax0=0;
dm(msg)=ax0;
dm(mcready)=ax0;
dm(bcctl)=ax0;
dm(bcnth)=ax0;
dm(nobgsub)=ax0;

cntr=0xc00;
i0=^outbuf0;
do clrmem until ce;
clrmem:      dm(i0,m2)=ax0;
i4=0x1e00;
i4=^workbuf;

/* let system stabilize */

cntr=1000;
do bpirq until ce;
cntr=16000;
do bpirq1 until ce;
bpirq1: nop;
bpirq: nop;

```

```

my1=io(0);      /* init for first readapds */
i1 = 0x3fe5;
dm(i1,m2)=0x0;  /*set pf0-7 to 0 */

call lightsoff;
ifc=0xff;       /*clear all pending irqs*/

jump mainbody;  /* for testing only */

jump waitmc;    /* wait for micro to init data areas */

/* INTERRUPT SERVICE ROUTINES-- */

irqfl:
my1=io(1);      /* read 0 setup 1 */
dm(i4,m5)=my1;
my1=io(2);      /* read 1 setup 2 */
dm(i4,m5)=my1;
my1=io(3);      /* read 2 setup 3 */

/* ***** for new board, change to io(4) below */
/* ***** for old board, change to io(3) below */
dm(i4,m5)=my1;
my1=io(4);      /* read 3 setup 4 */
dm(i4,m5)=my1;
my1=io(0);      /* read 4 setup 0 */
dm(i4,m5)=my1, af=af+1;

rti;

procnewctl:
rti;

doidle:
rti;

bdmadone:
rti;

tim_exp:
rti;

/* This loop is executed while NOT capturing. It only responds to 2 interrupts
IRQ 1 means that there is new data in the control area so reinit all variables
IRQ 0 means that there is new code in dm that must be moved to pm and executed
*/

Mainbody:
if flag_in call capture;
ax0=dm(lena);   /*turn on lasers */
ar=setbit 7 of ax0; /*enable overflow bit of a/d converter */
dm(pflags)=ar;

```

```

    imask = 0x206; /*allow idle or new ctrl msg */
    ena ints;
    nop;
    nop;
    nop;
    nop;
    i0 = ^workbuf;
    ax0 = dm(i0, m2); /* write background in same order as recbuf */
    dm(bkgnd) = ax0;

    ax0 = dm(i0, m2);
    dm(bkgnd+2) = ax0;

    ax0 = dm(i0, m2);
    dm(bkgnd+4) = ax0;

    ax0 = dm(i0, m2);
    dm(bkgnd+6) = ax0;

    ax0 = dm(i0, m2);
    dm(bkgnd+8) = ax0;

    jump mainbody;

/* All subroutines here at the end */

capture:
    i4 = 0x1e00;
    i5 = 0x1e00;
    ax0 = dm(i4); /*turn on lasers */
    ar = clrbit 7 of ax0; /* disable overflow bit of a/d convt */
    dm(pflags) = ar;
    dis ints;
    call clrhst;
    af = pass 0;
    i4 = ^workbuf;
    i3 = ^hist;
    i5 = ^workbuf;
    i7 = ^x8bkgnd;
    imask = 0x200;
    ena ints;

    cntr = 4; /* do four background histograms */
    do bigloop until ce;

    cntr = 4000; /* take 4000 samples for histogram */
    do bldhist until ce;
pace:  ar = pass af; /* any samples to proc */
    if le jump pace;
    ax0 = dm(i5, m6); /* get samp */
    ar = ax0 - 0x1000; /* range check */
    if ge jump bldhist;
    ay0 = ^hist; /* base addr */

```

```

    ar=ax0+ay0;      /* base + offset */
    i3=ar;           /*point into hist */
    ay1=dm(i3,m3);   /* get current val */
    ar=ay1+1;
    dm(i3,m3)=ar;
bldhist:  nop;

    /* now find most frequent occurrence in hist */

    i3=^hist;
    cntr=0xff;
    ax0=dm(i3,m2);   /* get first entry */
    dm(svi3)=i3;
    ay0=ax0;          /* ay0 holds highest val so far */

    do findmax until ce;
    ax0=dm(i3,m2);
    ar=ax0-ay0;
    if le jump findmax;
    ay0=ax0;          /*new high val found */
    dm(svi3)=i3;      /* save offset +1 (post inc) */
findmax:  nop;
    ax0=dm(svi3);     /* get ptr to best +1 */
    ay0=^hist + 1;    /* since i3 is plus 1 */
    ar=ax0-ay0;       /*compensate for base */
    si=ar;            /* save bkgnd for shift later */
    ay0=^hist;
    ar=ar+ay0;        /*calc index to peak */

    i3=ar;            /*point to peak in hist */
    ay0=dm(i3,m1);    /*get pk, pnt to previous */
    ax0=dm(i3,m0);    /*get prev, pnt to other side of pk */
    ay1=dm(i3,m1);    /* pnt back to peak */
    ax1=1;
    ar=ax0-ay1;       /* which pk is second highest */
    if lt jump poside;
    ax1=-1;           /* int pk will be less */
    ay1=ax0;

poside:      /* now ay1 has val of 2nd peak */
    sr=lshift si by 3 (lo); /* mult by 8 */
    si=ay0;
    ay0=sr0;          /*ay0 is 8x bkgnd and is ready for adj */
    sr=lshift si by -2(lo); /* get 1/4 of peak height */
intlp:  ar=ay1-sr0;    /* reduce 2nd pk by 1/4 of main */
    if lt jump intend;
    ay1=ar;
    ar=ax1+ay0;       /*adjust background + or - */
    ay0=ar;
    jump intlp;
intend:  dm(i7,m5)=ay0; /* save final bkgnd */

    call clrhst;
bigloop:  modify(i5,m5); /*point to next channel */

    i3=^x8bkgnd;
    i1=^bkgnd;
    cntr=4;
    do div8 until ce;
    si=dm(i3,m2);     /*get avg background , inc +2*/
    sr=lshift si by -3 (lo);
div8:  dm(i1,m0)=sr0;

```

```

    af=pass 0;
    ax0=0;
    dm(misscnt)=ax0;
    dm(buf0cnt)=ax0;
    dm(buf1cnt)=ax0;
    dm(otrcnt)=ax0;
    dm(actbuf)=ax0;
    ax0=dm(clock);
    dm(0x3ff1)=ax0; /* set sample clk freq */
    i4=^workbuf;
    i5=^workbuf;
    ifc=0xff;
    imask=0x200; /* ena fl ints */
    i0=^outbuf0;
resetc: if not flag_in jump capexit;
    ax0=0;
    ar=pass af;
    dm(afreg)=ar;
    ena ints;
newsamp: if not flag_in jump capexit;
    ar=pass af; /* load ax0 with # of new samps */
    ar=ar-64; /* are we 64 records ahead */
    if le jump newsamp;
    af=af-1, ay0=dm(i5,m6); /* get ssc, inc to next record */
    ax0=dm(thresh);
    ar=ax0-ay0; /* above thresh then its a bead */
    if ge jump newsamp;

    /* got a bead */

    dm(pki5)=i5;
    dm(fstart)=i5;
    ax1=ay0; /* init peak ssc */
    ay1=0; /* init width count */

moresamp:
    af=af-1, ay0=dm(i5,m6); /* next next ssc, dec rec count */
    ar=ay1+1; /* inc width count */
    ay1=ar, ar=ax1-ay0; /* is new ssc higher */
    if ge jump ckdone;
    dm(pki5)=i5;
    ax1=ay0; /* save peak ssc */
ckdone: ar=pass af; /* out of records */
    if eq jump bubble; /* if 0 then bubble */
    ar=ax0-ay0; /* back below thresh yet? */
    if le jump moresamp;
    i6=dm(pki5);
    modify(i6,m4); /* since post inc, sub 20 */
    dm(wid)=ay1;

    /* build a record with peaks or estimates of peaks */
    i2=0x1e00;
    i2=i6;
    call i2sum;
    dm(recbuf)=ay1; /* store ssc in recbuf */

    modify(i6,m5); /* add 1 to point to fl2 pk -15 */
    i2=i6;
    call i2sum;
    dm(recbuf+1)=ay1; /* store cl1 sum */

```

```

modify(i6,m5); /* add 1 to point to f13 pk -15 */
i2=i6;
call i2sum;
dm(recbuf+2)=ay1; /* store ci2 sum */

modify(i6,m5); /* add 1 to point to f14 pk -15 */
i2=i6;
call i2sum;
dm(recbuf+3)=ay1; /* store ci3 sum */

modify(i6,m5); /* add 1 to point to f11 pk -15 */
dm(svi6)=i6; /*align point with red laser */
modify(i6,m4); /* back up 20 more to f11 peak -35 */
modify(i6,m4); /* back up 20 more to f11 peak -55 */
/*modify(i6,m4); back up 20 more to f11 peak -75 */

/* since f11 is a different laser, refind peak to allow for alignment
differences, search from -15 to +16 */

ax0=dm(i6,m6); /* get 1st f11 sample */
m7=160;
m7=350; /*for new filter at 50 khz */
modify(i6,m7);
m7=0;
ay0=dm(i6,m6);
ar=ax0+ay0;
sr=lshift ar by -2 (LO);
ay0=sr0;
ar=ar+ay0;
dm(x8bkgnd+4)=ar;

i6=dm(svi6); /* test using no peak find */
m7=40;
modify(i6,m7); /*move right 8 samples due to skew of 50k filter */
m7=0;

cntr=7;
modify(i6,m6);
dm(pki5)=i6;
ay0=dm(i6,m6); /* get 1st f11 sample */
ax1=ay0; /* move to old f11 val */
do rp1pk until ce;
ay0=dm(i6,m6);
ar=ax1-ay0; /* is it a new peak? */
if gt jump rp1pk;
dm(pki5)=i6; /* store addr +5 of new peak (post inc) */
ax1=ay0;
rp1pk: nop;
i6=dm(pki5);
modify(i6,m4); /* sub 20 to back up 3 samples from pk (was +5) */
modify(i6,m4); /* sub 20 to back up 4 samples 50 k version */
i2=i6;
call i2sum14;
sr=lshift si by -3 (HI); /* shift carry */
sr=sr or lshift ar by -3 (LO);
dm(recbuf+4)=sr0; /*store rp1 */
l2=0; /* restore for non circ buffr use */
/* ok, we have singlet and recbuf is built */

i7=^recbuf;
i3=^x8bkgnd;
cntr=4;

```

```

do subback until ce;
ax0=dm(i7,m7); /* get val inc 0 */
ay0=dm(i3,m2);
ar=ax0-ay0;
if lt ar=pass 0;
subback:    dm(i7,m5)=ar;
            ax0=dm(nobgsub);
            ar=pass ax0;
            if ne jump recbdone;
            ax0=dm(i7,m7); /* get val inc 0 */
            ay0=dm(i3,m2);
            ar=ax0-ay0;
            if lt ar=pass 0;
            dm(i7,m5)=ar;
            /******Sub r/t bg from rp1******/

recbdone:

            /* record bead count */
            ay1=dm(bcctl);
            ar=ay1+1;
            dm(bcctl)=ar;
            ar=dm(bccth);
            ar=ar+C;
            dm(bccth)=ar;

            i2=^recbuf;
            ay1=dm(actbuf);
            none=pass ay1;
            if ne jump usebuf1;

            ay1=dm(buf0cnt);
            none=pass ay1;
            if gt jump noi0init; /* is this first of block */
            i0=^outbuf0;

noi0init:
            ax1=dm(buf0siz); /* check for overflow */
            none = ax1 - ay1;
            if ne jump noover;
            ay1=dm(misscnt);
            ar=ay1+1;
            dm(misscnt)=ar;
            jump resetc;
noover:    cntr=5;
            do movout0 until ce;
            ar=dm(i2,m2);
movout0:    dm(i0,m0)=ar; /* store ssc inc by 2 for even addr*/

            ay0=dm(buf0cnt);
            ar=ay0+1;
            dm(buf0cnt)=ar;
            ay0=dm(buf0siz); /* enough records to irq 167? */
            ar=ar-ay0;
            if ne jump resetc;
            set fl0;
            nop; /* c167 needs 50 ns min to get edge */
            nop;
            reset fl0;
            ay1=1;
            dm(actbuf)=ay1; /* change to buffer 1 */

```

```

        jump resetc;

usebuf1:
    ay1=dm(buf1cnt);
    none=pass ay1;
    if gt jump noi1init;      /* is this first of block */
    i0=^outbuf1;

noi1init:
    ax1=dm(buf1siz);          /* check for overflow */
    none = ax1 - ay1;
    if ne jump noover1;
    ay1=dm(misscnt);
    ar=ay1+1;
    dm(misscnt)=ar;
    jump capture;             /* major overflow reset and restart capture */
noover1:    cntr=5;
    do movout1 until ce;
    ar=dm(i2,m2);
movout1:    dm(i0,m0)=ar;      /* store ssc inc by 2 for even addr*/

    ay0=dm(buf1cnt);
    ar=ay0+1;
    dm(buf1cnt)=ar;
    ay0=dm(buf1siz);          /* enough records to irq 167? */
    ar=ar-ay0;
    if ne jump resetc;
    set fl1;
    nop;                       /* c167 needs 50 ns min to get edge */
    nop;
    reset fl1;
    ay1=0;
    dm(actbuf)=ay1; /* change back to buffer 0 */
    jump resetc;

capexit:
    i1 = 0x3fe5;
    dm(i1,m2)=0x0;             /*set pf0-7 to 0 */

    ifc=0xff;
    rts;

i2sum14:
    m0=5;
    ar=pass 0,ay1=dm(i2,m0);
    si=0;                       /* use for carry */
    dm(svcry)=si;
    cntr=20;
    do grate4 until ce;
    ar=ar+ay1,ay1=dm(i2,m0);
    if not ac jump grate4;
    dis ints;
    ena sec_reg;
    ay0=dm(svcry);
    ar=ay0+1;
    dm(svcry)=ar;
    dis sec_reg;
    ena ints;
grate4:    nop;
    ay1=ar;
    si=dm(svcry);
    m0=2;
    rts;

```



```

i2sum:  m0=5;
        ar=pass 0,ay1=dm(i2,m0);
        cntr=8;
        do grate until ce;
grate:  ar=ar+ay1,ay1=dm(i2,m0);
        ay1=ar;
        m0=2;
        rts;

bubble:
        if not flag_in jump capexit;
        ar=pass af;          /* load ax0 with # of new samps */
        ar=ar-256;          /* are we 64 records *4 ahead */
        if le jump bubble;
        af=af-1,ay0=dm(i5,m6); /*get ssc, inc to next record */
        ax0=dm(thresh);
        ar=ax0-ay0;          /* are we below thresh yet */
        if le jump bubble;    /* no, keep waiting */
        jump newsamp;

lightsoff:
        ax0=0x24d;
        ax0=0x63;          /******test*****/
        dm(version)=ax0;
        rts;

clrhist:
        i3=^hist;
        ax0=0;
        cntr=0x1000;
        do zit until ce;
zit:    dm(i3,m2)=ax0;
        rts;

waitmc:
        toggle fo;
        ax0=dm(mcready);    /* wait until mc writes non zero to mcready */
        ar=pass ax0;
        if eq jump waitmc;
        jump mainbody;

.ENDMOD;

```

## INTERNATIONAL SEARCH REPORT

 International application No.  
 PCT/US99/10318

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 15/20

US CL : 702/45

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 702/45, 702/49, 73/861, 436/63

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched  
NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

U.S. PTO APS

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,448,706 A (Fleming et al.) 05 September 1995, (05/09/95) Abstract, line 3; col. 4, lines 32-38; col. 8, lines 56-65; col. 20, lines 9-22 and lines 34-49; col. 1, line 10, lines 25-26, line 40, line 53, line 66	1,20,22,24,26
X,P	US 5,880,474 A (Norton et al.) 09 March 1999, (09/03/99) col. 1, line 12; col. 2, lines 5-34; col. 4, lines 38-49.	1,20,22,24,26,2,2 1,23,25,27
A	US 5,299,141 A (Hungerford et al.) 29 March 1994, (19/03/94) see entire document.	1-27
A	US 4,478,088 A (Loveland) 23 October 1984, (23/10/84) see entire document.	1-27
A,E	US 5,909,278 A (Deka et al.) 01 June 1999, (01/06/99) see entire document.	1-27



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
*A* document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
*E* earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*G* document member of the same patent family
*O* document referring to an oral disclosure, use, exhibition or other means	
*P* document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

06 AUGUST 1999

Date of mailing of the international search report

30 AUG 1999

 Name and mailing address of the ISA/US  
 Commissioner of Patents and Trademarks  
 Box PCT  
 Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

LINH M. NGUYEN

Telephone No. (703) 308-0956

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US99/10318

## C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,734,422 A (Maurer et al.) 31 March 1998, (31/03/98) col.3, line 27.	18, 19